

OSCAL-based AI-augmented CISO Agent

Anca Sailer, Yuji Watanabe, Takumi Yanagawa,
Hirokuni Kitahara, Saki Takano,
IBM Research

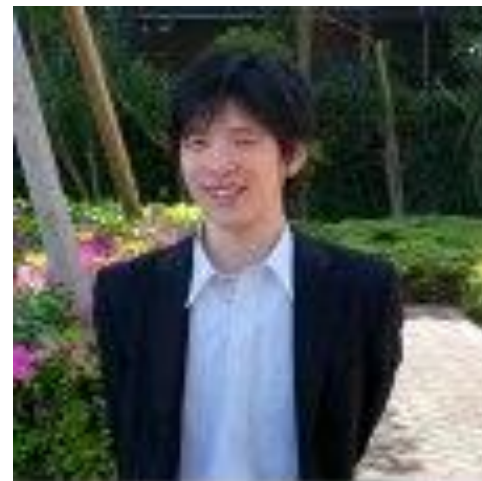
March 19, 2025



Anca Sailer



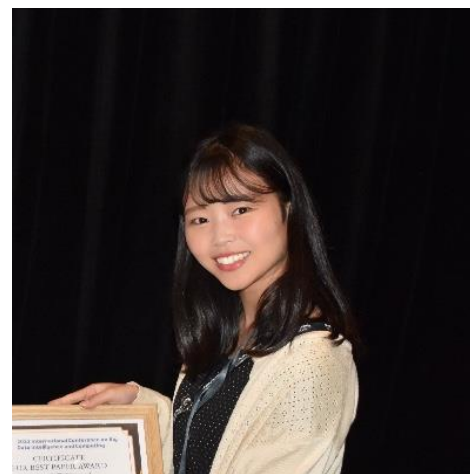
Yuji Watanabe



Takumi Yanagawa



Hirokuni Kitahara



Saki Takano

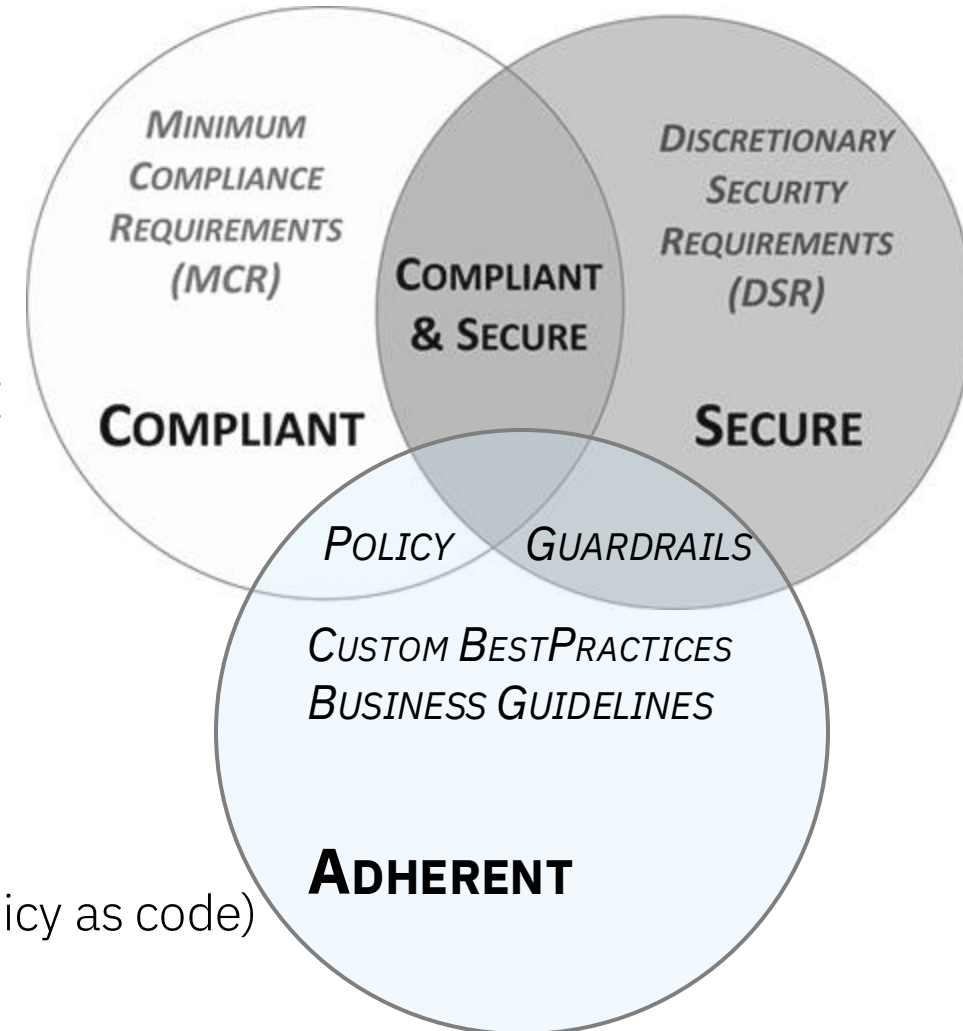
Compliance, Adherence, Policy

COMPLIANCE (enforcement)

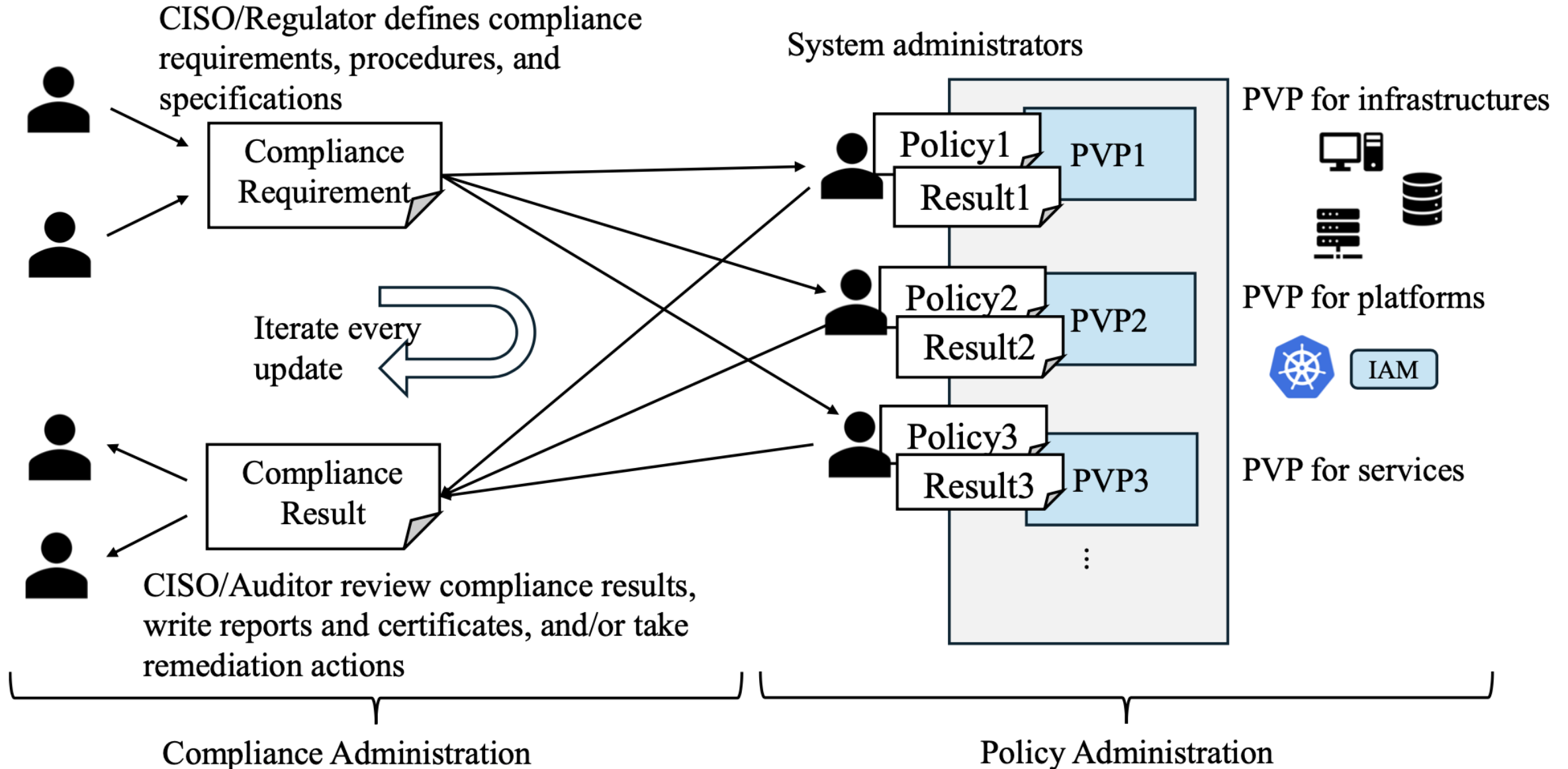
- Best Practices become Policy Requirements
- Continuous compliance (enforcement) requires policy development for task specific automation
- Applies at any level (infrastr., data, AI, app, process) [Compliance4AI](#)
- Applies for any domain/app (security, business, finance)

COMPLIANCE LIFE-CYCLE (all phases benefit from [AI4Compliance](#))

1. Define and author controls (compliance as code)
2. Implement controls (tech, app specific rules)
3. Assess controls and enforce/provide results (tech, app specific; policy as code)
4. Audit and report



Compliance-as-Code vs policy-as-Code Administration



OSS: OSCAL, Trestle, Agile Authoring, Compliance-to-Policy

<https://pages.nist.gov/OSCAL/>

<https://github.com/oscal-compass>

<https://github.com/oscal-compass/compliance-trestle>

<https://oscal-compass.github.io/compliance-trestle/>



OSCAL is a NIST framework & language for managing compliance artifacts as code end-to-end

From selection of security controls through implementation and assessment

To plans of actions for remediations and mitigation



TRESTLE is an opinionated implementation of the OSCAL standard

Allows editing and manipulation of OSCAL documents while making sure the schemas are enforced

Provides an SDK



AGILE AUTHORIZING is a collaborative platform enabling various compliance personas to orchestrate their individual aspects of the compliance artifacts via an interface of their choice

Trestle-based GitOps automated workflow
Ensures artifacts consistency and traceability



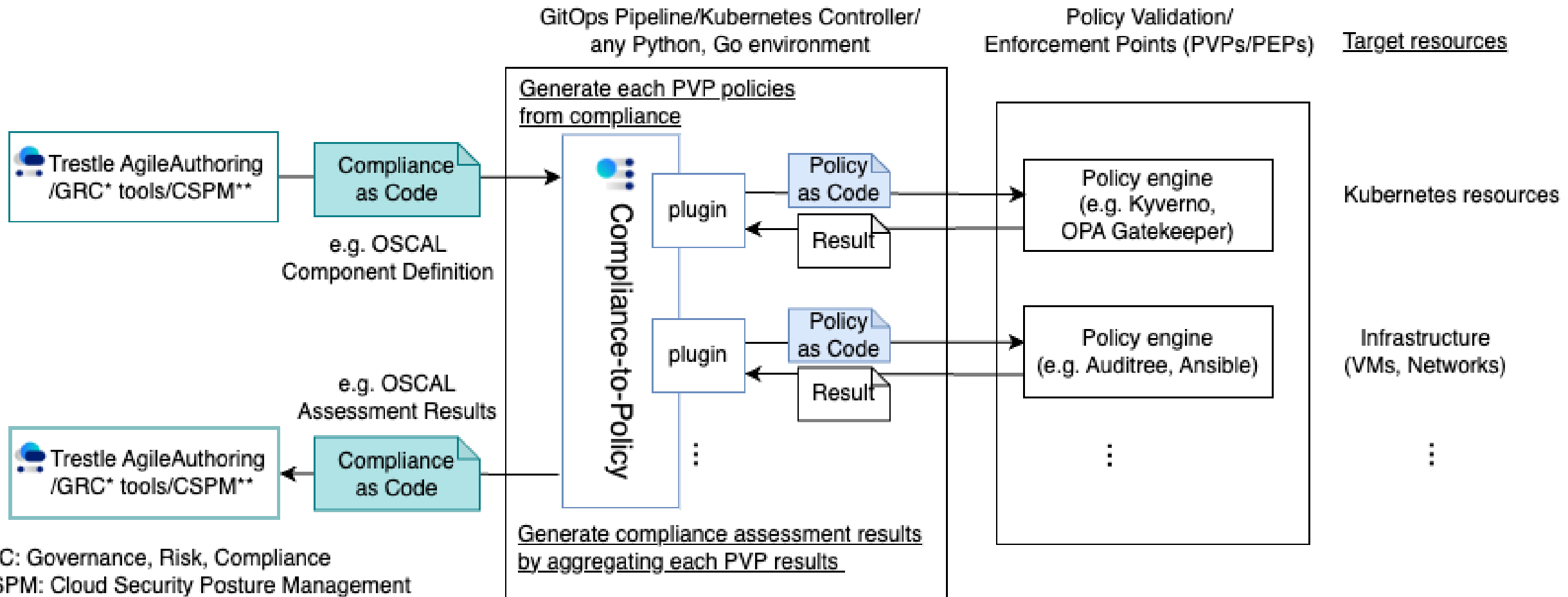
COMPLIANCE_TO_POLICY is a GitOps extension as a pluggable bridge to normalize the policy administration in the policy validation tools

Bridge between compliance-as-code and policy-as-code

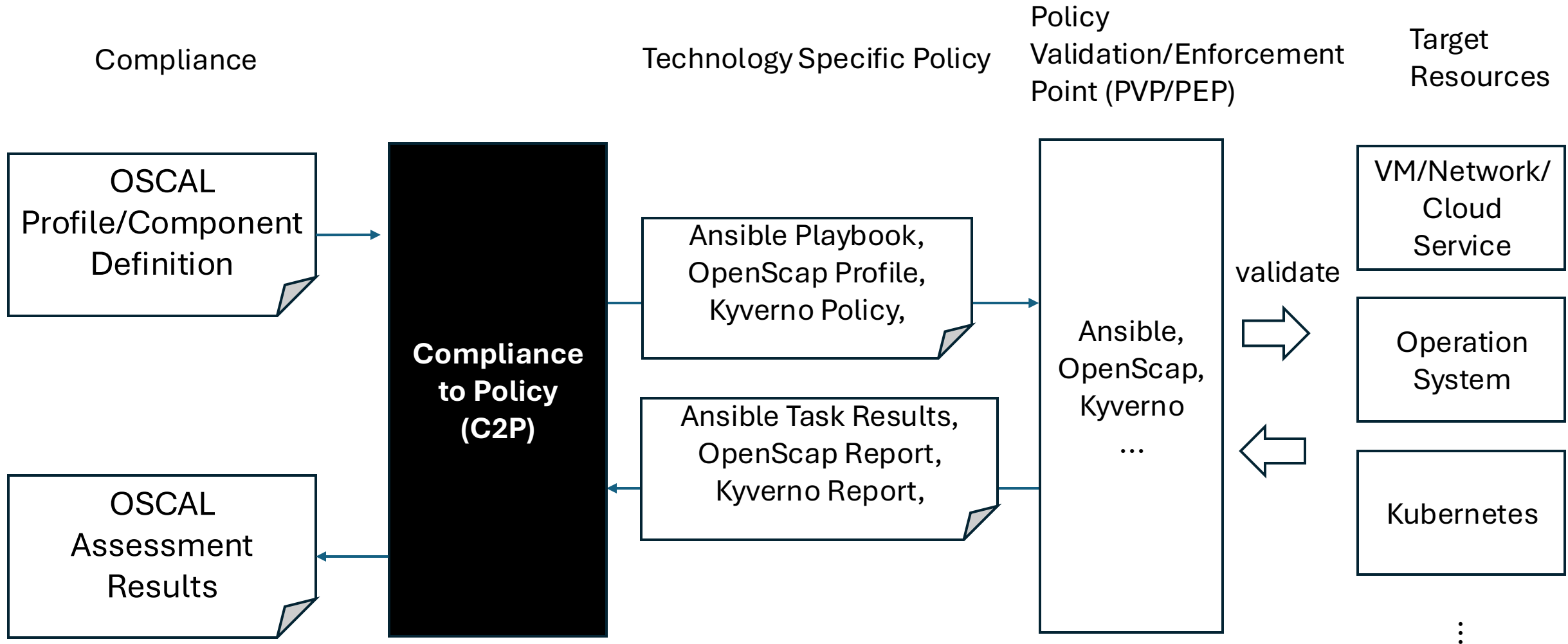
Compliance-to-Policy (C2P) and plugin architecture



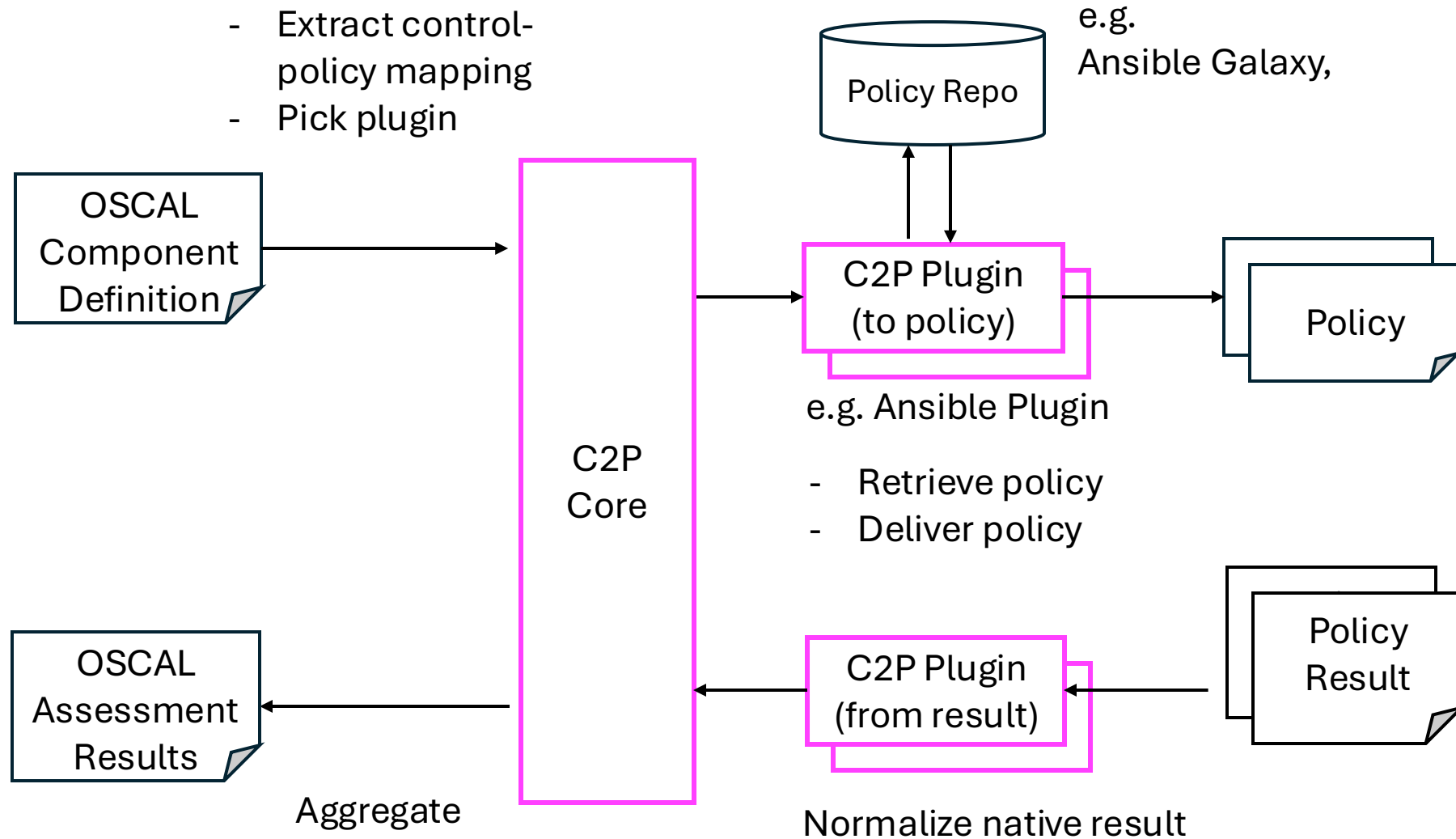
Flexibility in choice of policy engines and compliance framework
Community-driven plugin extension



Conceptual Diagram of C2P

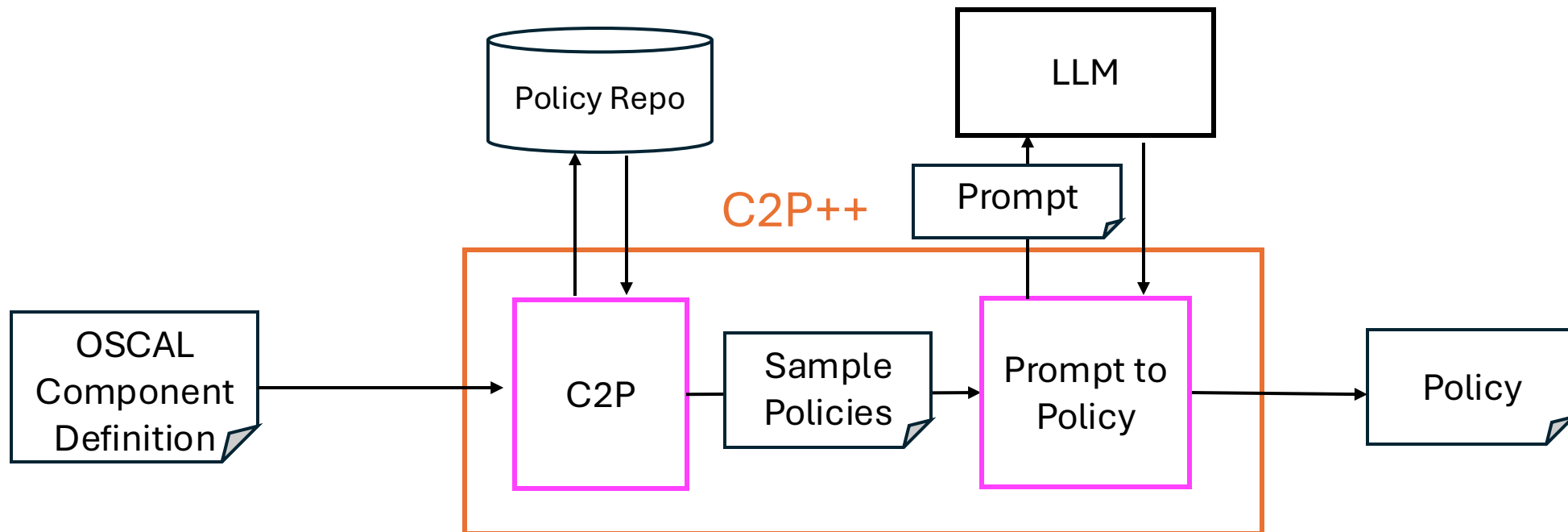


C2P Policy Generation Process

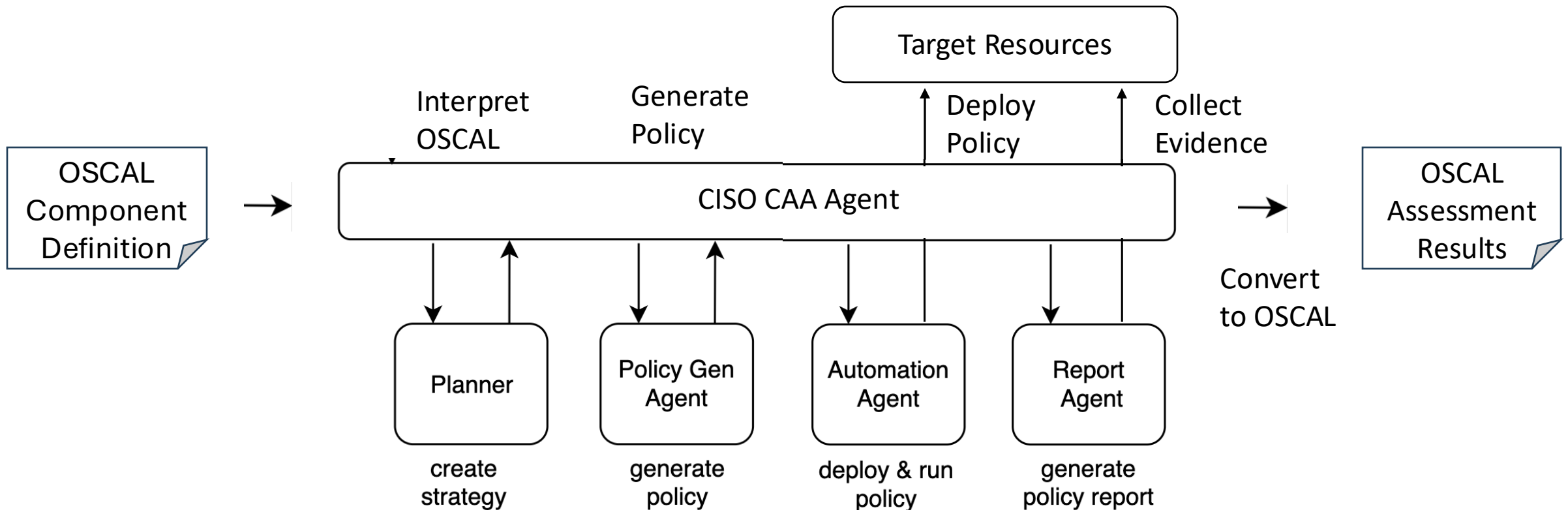
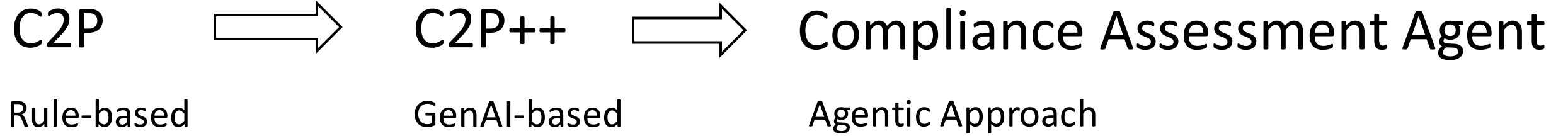


GenAI-based C2P (C2P++)

- C2P enables an end-to-end OSCAL flow with various PVPs.
- However, implementing new policies for new compliance controls remains a **human-intensive** task.
- Use GenAI to **fully automate** policy generation



Transition to AI Agent for Full-Automation



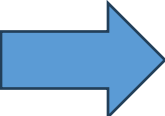
What is CISO Compliance Assessment Agent (CAA)? Why?

Task example in Compliance Assessment

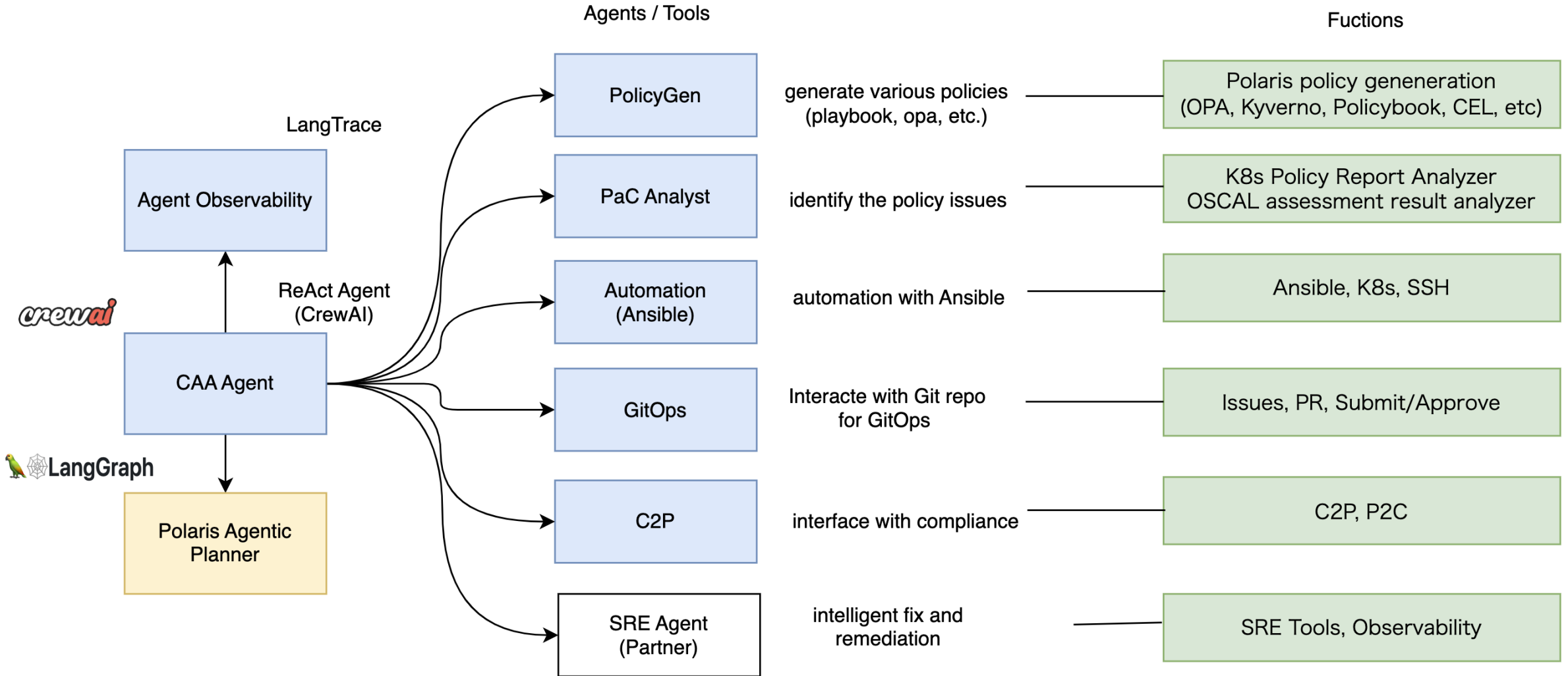
Given new compliance check requirement,

- generate new policy (code), deploy it, enable the automated check,*
- collect evidences, and then report compliance posture.*

Issues in compliance assessment today

- many manual efforts
 - risk of inconsistency, non-compliance
 - automation for fixed goal with hard wiring
 - less agility
 - multiple persona with different knowledge and experience
 - persona: Compliance person, System admin, Auditor / CISO
 - knowledge: compliance requirement, system architecture, policy engine, data and APIs
- 
- CISO CAA Agent**

CISO Compliance Assessment Agent (CAA)

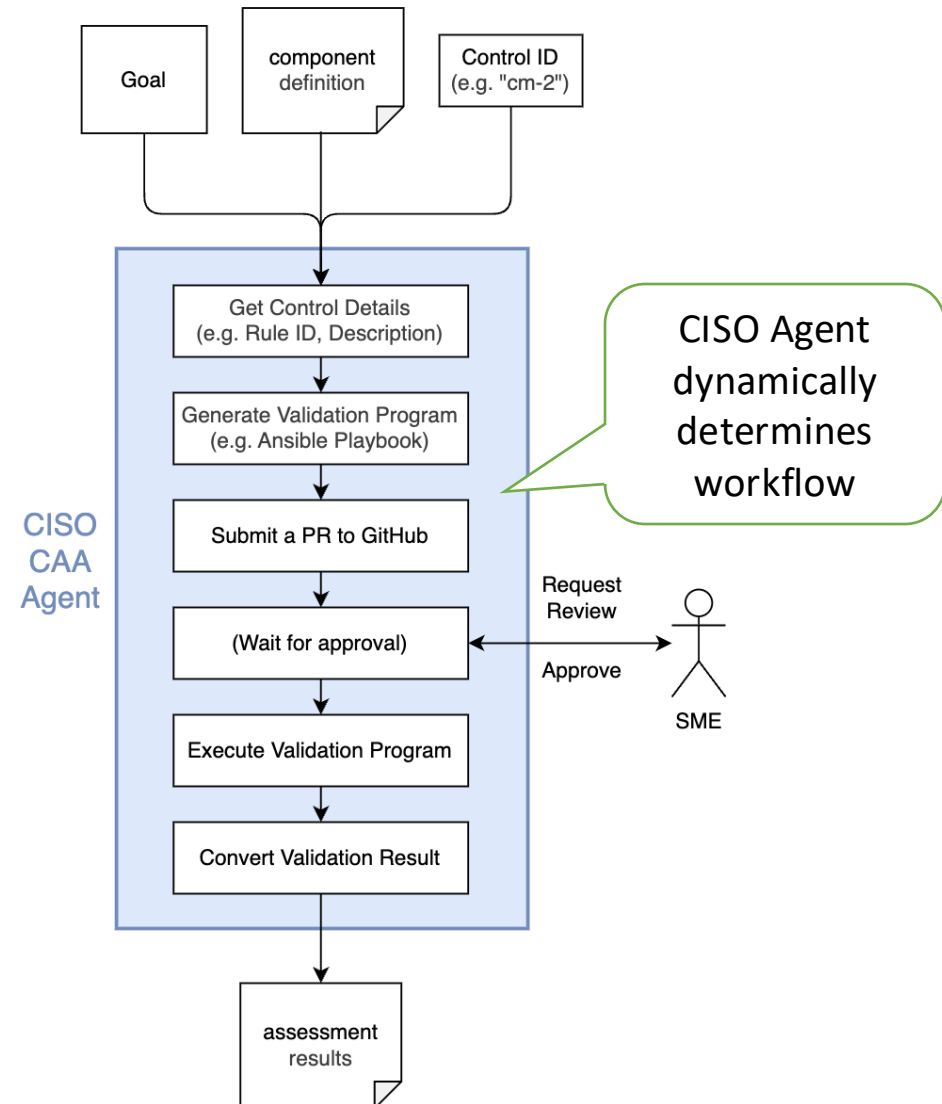


CISO Agent for Compliance Automation with OSCAL Artifacts

CISO Agent goal description

```
role: >
  CISO CAA Agent
goal: >
  As a Compliance Assessment Engineer, given a new compliance rule recorded
  in the OSCAL Component Definition:
  - identify only the relevant rule details (rule_id, rule_description) and
  validation tool (Ansible or not) from the OSCAL Component Definition that are
  related to the given compliance requirement. From the Validation Component
  (component.type = "Validation"), retrieve the corresponding check methods
  (Check_Id) for the relevant rules.
  - assess the target system and produce OSCAL Assessment Results
  - for that, runs a code on an appropriate check tool (currently, only
  Ansible is supported validation tool. report error if it is not Ansible.)
  - the code is generated using some functions with the rule description.
  similar example can be used to improve the code.
  - the code should be posted to Github as PR. the PR must be approved by
  SME. The code should not be run before SME's approval.
  - finally, please push the OSCAL Assessment Results to github repository
```

CISO Agent workflow



OSCAL Artifacts (Inputs / Output) for CISO Agent

OSCAL Component Definition

```

"control-implementations": [
  {
    "uuid": "34492384-b711-4757-9c75-bb146fa4390e",
    "source": "https://github.com/usnistgov/oscal-content/blob/main/nist.gov/SP800-53A/Control-Implementations/Control-Implementations-2020-08-14.yaml",
    "description": "NIST Special Publication 800-53 Revision 5 HIGH IMPACT BASELINE",
    "implemented-requirements": [{
      "uuid": "0ef038f6-aa2d-4339-a827-da20c54bb9fd",
      "control-id": "cm-2",
      "description": "Ensure SSH access is restricted through connection limits",
      "props": [{
        "name": "Rule_Id",
        "ns": "http://github.com/oscal-compass/schemas/oscal/cd",
        "value": "ssh_maxstart_up",
        "name": "Rule_Description",
        "ns": "http://github.com/oscal-compass/schemas/oscal/cd",
        "value": "SSH MaxStartups is set to limit simultaneous connections.",
        "remarks": "rule_set_0"
      }],
      "name": "Rule_Id",
      "ns": "http://github.com/oscal-compass/schemas/oscal/cd",
      "value": "ssh_maxstart_up",
      "name": "Rule_Description",
      "ns": "http://github.com/oscal-compass/schemas/oscal/cd",
      "value": "SSH MaxStartups is set to limit simultaneous connections.",
      "remarks": "rule_set_0"
    }],
    "name": "Rule_Id",
    "ns": "http://github.com/oscal-compass/schemas/oscal/cd",
    "value": "ssh_maxstart_up",
    "name": "Rule_Description",
    "ns": "http://github.com/oscal-compass/schemas/oscal/cd",
    "value": "SSH MaxStartups is set to limit simultaneous connections.",
    "remarks": "rule_set_0"
  }],
  {
    "uuid": "0ef038f6-3401-4bbe-9bc2-11627ad6563f",
    "control-id": "ac-10",
    "description": "Ensure SSH session concurrency is restricted.",
    "props": [{
      "name": "Rule_Id",

```

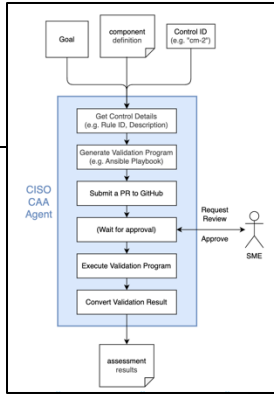
OSCAL Assessment Results

```

"subjects": [
  {
    "subject-uuid": "2961086a-57d5-465c-a209-d94db41a9a61",
    "type": "inventory-item",
    "title": "Ansible Playbook: check_ssh_maxstartup",
    "props": [{
      "name": "resource-id",
      "value": "check_ssh_maxstartup"
    }],
    {
      "name": "rule_id",
      "value": "ssh_maxstart_up"
    },
    {
      "name": "result",
      "value": "failure"
    },
    {
      "name": "evaluated-on",
      "value": "2025-03-17T07:12:34+00:00"
    },
    {
      "name": "reason",
      "value": "\nPLAY [Check if SSH MaxStartups is Configured] ***
[Gathering Facts] *****\nok:
configured] *****\nfatal: [localhost]: FAILED! :
\"grep\\\", \\\"-E\\\", \\\"^MaxStartups\\\", \\\"/etc/ssh/sshd_config\\\", \\\"c
\\\"2024-07-17 03:34:40.302900\\\", \\\"failed_when_result\\\": true, \\\"msg\\\"
\\\"start\\\": \\\"2024-07-17 03:34:40.296682\\\", \\\"stderr\\\": \\\"grep: /etc,
\\\", \\\"stderr_lines\\\": [\\\"grep: /etc/ssh/sshd_config: No such file or direct
\\\"stdout_lines\\\": []}\n\nPLAY RECAP *****
: ok=1  changed=0  unreachable=0  failed=1  skipped=0  rescued=0  igno

```

Assessment Result of the Rule ID: "ssh_maxstart_up"



Benchmarking

URL: <https://arxiv.org/abs/2502.05352>

arXiv > cs > arXiv:2502.05352 Search... Help | Advan

Computer Science > Artificial Intelligence

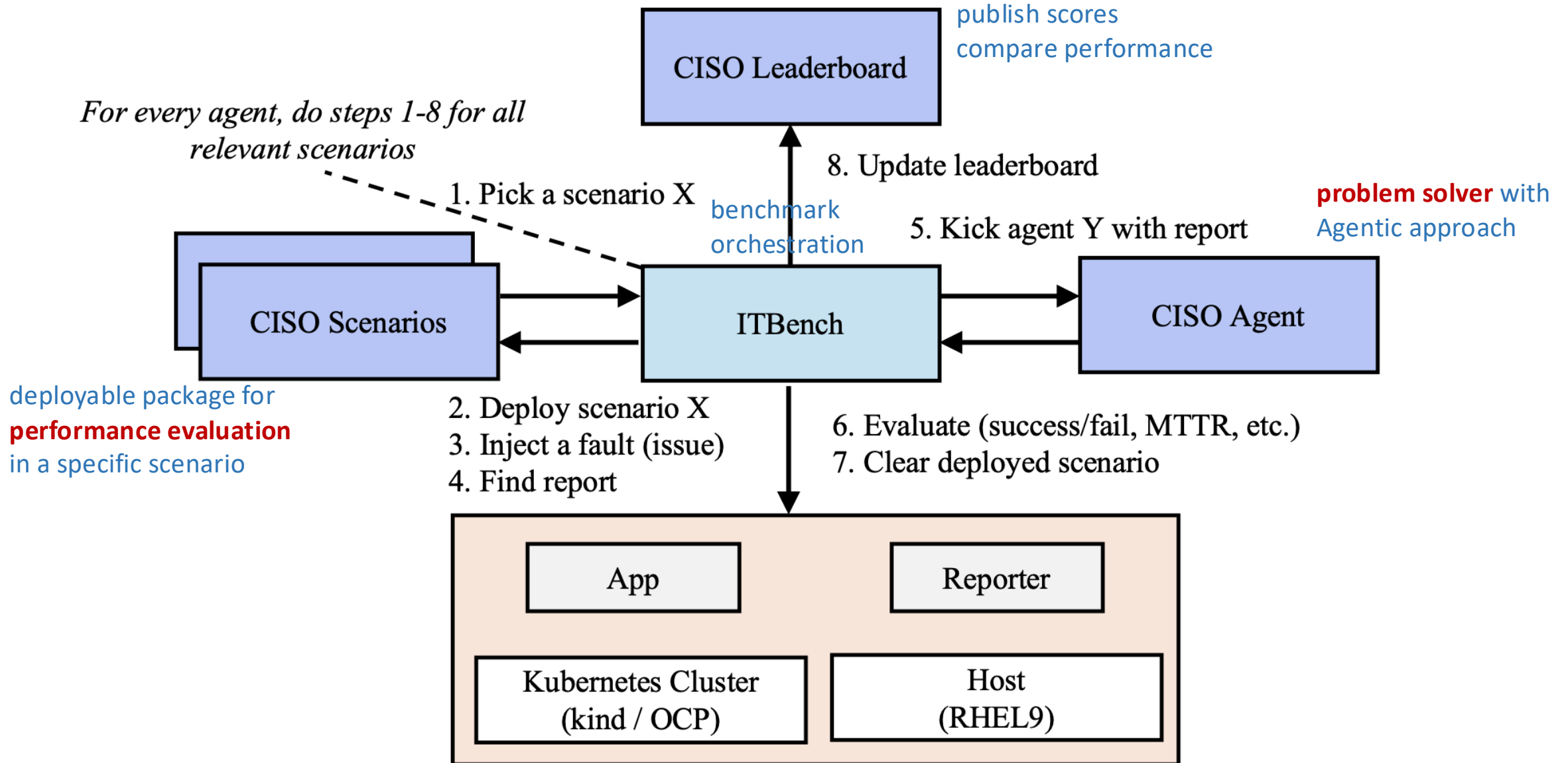
[Submitted on 7 Feb 2025]

ITBench: Evaluating AI Agents across Diverse Real-World IT Automation Tasks

Saurabh Jha (1), Rohan Arora (1), Yuji Watanabe (1), Takumi Yanagawa (1), Yinfang Chen (2), Jackson Clark (2), Bhavya Bhavya (1), Mudit Verma (1), Harshit Kumar (1), Hirokuni Kitahara (1), Noah Zheutlin (1), Saki Takano (1), Divya Pathak (1), Felix George (1), Xinbo Wu (2), Bekir O. Turkkan (1), Gerard Vanloo (1), Michael Nidd (1), Ting Dai (1), Oishik Chatterjee (1), Pranjal Gupta (1), Suranjana Samanta (1), Pooja Aggarwal (1), Rong Lee (1), Pavankumar Murali (1), Jae-wook Ahn (1), Debanjana Kar (1), Ameet Rahane (1), Carlos Fonseca (1), Amit Paradkar (1), Yu Deng (1), Pratibha Moogi (1), Prateeti Mohapatra (1), Naoki Abe (1), Chandrasekhar Narayanaswami (1), Tianyin Xu (2), Lav R. Varshney (2), Ruchi Mahindru (1), Anca Sailer (1), Laura Shwartz (1), Daby Sow (1), Nicholas C. M. Fuller (1), Ruchir Puri (1) ((1) IBM, (2) University of Illinois at Urbana-Champaign)

Realizing the vision of using AI agents to automate critical IT tasks depends on the ability to measure and understand effectiveness of proposed solutions. We introduce ITBench, a framework that offers a systematic methodology for benchmarking AI agents to address real-world IT automation tasks. Our initial release targets three key areas: Site Reliability Engineering (SRE), Compliance and Security Operations (CISO), and Financial Operations (FinOps). The design enables AI researchers to understand the challenges and opportunities of AI agents for IT automation with push-button workflows and interpretable metrics. ITBench includes an initial set of 94 real-world scenarios, which can be easily extended by community contributions. Our results show that agents powered by state-of-the-art models resolve only 13.8% of SRE scenarios, 25.2% of CISO scenarios, and 0% of FinOps scenarios. We expect ITBench to be a key enabler of AI-driven IT automation that is correct, safe, and fast.

End-to-end Benchmarking Process for CISO Agent



CISO Benchmark Scenarios

- 50 scenarios implemented from 4 different scenario types
- Covering various compliance assessment situations
 - different environment (K8s, OpenShift, RHEL9 hosts)
 - different policy engines (Kyverno, OPA, Ansible, Kubectl)
 - 1 policy or 2 different policies (e.g. fetch + check)
 - different automation goals

Simple

Scenario Type	Env / CIS-b	Policy	Agent Goal	Num
Gen-CIS-b-K8s-Kyverno	K8s	1 (Kyverno)	Generate + Deploy + Check	10
Gen-CIS-b-K8s-Kubectl-OPA	K8s	2 (OPA Rego, Kubectl)	Generate + Fetch + Check	10
Gen-CIS-b-RHEL9-Ansible-OPA	RHEL9	2 (OPA Rego, Ansible)	Generate + Fetch + Check	20
Upd-CIS-b-K8s-Kyverno	K8s	1 (Kyverno)	Update + Redeploy + Check	10

Complex

Example

Goal

Ensure cron is restricted to authorized users. The system must: Allow only authorized users to schedule jobs using cron. Specific requirements:

- The file `/etc/cron.allow` must exist and specify the list of authorized users.
- The file `/etc/cron.deny` must not exist.
- Unauthorized users (e.g., `user1`, `user2`) must not be listed in `/etc/cron.allow`.

To check the condition, do the following steps. - collect configuration from the RHEL host and save it locally. you can use `ansible-playbook` to do that. - check if the condition is met by using `rego` policy with the input given by the step above.

for those steps, you need to create `ansible` playbook 'playbook.yml' and `OPA` `rego` policy 'policy.rego'. If you can fix the generated code, do it and run the fixed code again. Once you get a final answer, you can quit the work.

Trajectory

Thought: The task requires checking a `Rego` policy for a given input file. The policy and input file need to be prepared first. The goal is to ensure `cron` is restricted to authorized users on a `RHEL` host. To begin, an `Ansible` playbook is needed to collect the necessary configuration from the `RHEL` host.

Action: using tool `GeneratePlaybookTool`
Input: {"sentence": "Collect cron allow and deny configuration from RHEL host..."}

Observation:
- name: Collect Cron Allow and Deny Configuration
...
content: "{{ collected_data['content'] | b64decode | quote }}\n{{ cron_deny.stdout | quote }}"
dest: collected_data.json

Thought: Now that I have the playbook generated, I need to run it to collect the configuration from the `RHEL` host.

Action: using tool `RunPlaybookTool`
Input: {"playbook_file": "/tmp/agent/20250127171401/playbook.yml", ...}

Observation:
PLAY [Collect Cron Allow and Deny]
...
rhel9 : ok=6 changed=4
unreachable=0 failed=0 skipped=0 rescued=0
ignored=1

Thought: The playbook has been run successfully, and the configuration has been collected. Next, we need to generate an `OPA` `Rego` policy to check if the collected configuration meets the requirements.

Action: using tool `GenerateOPAREgoTool`
Input: {"sentence": "Ensure cron is restricted to authorized users", "input_file": "/tmp/agent/20250127171401/collected_data.json" ...}

Observation:
The generated policy is below:
package check
...
not contains(input, "root")
}
This policy file has been saved at
/tmp/agent/20250127171401/policy.rego.

Thought: Now that we have the `Rego` policy, we can use it to check the collected configuration.

Report filepaths that are created in the previous task.

Benchmarking CISO agent with various models

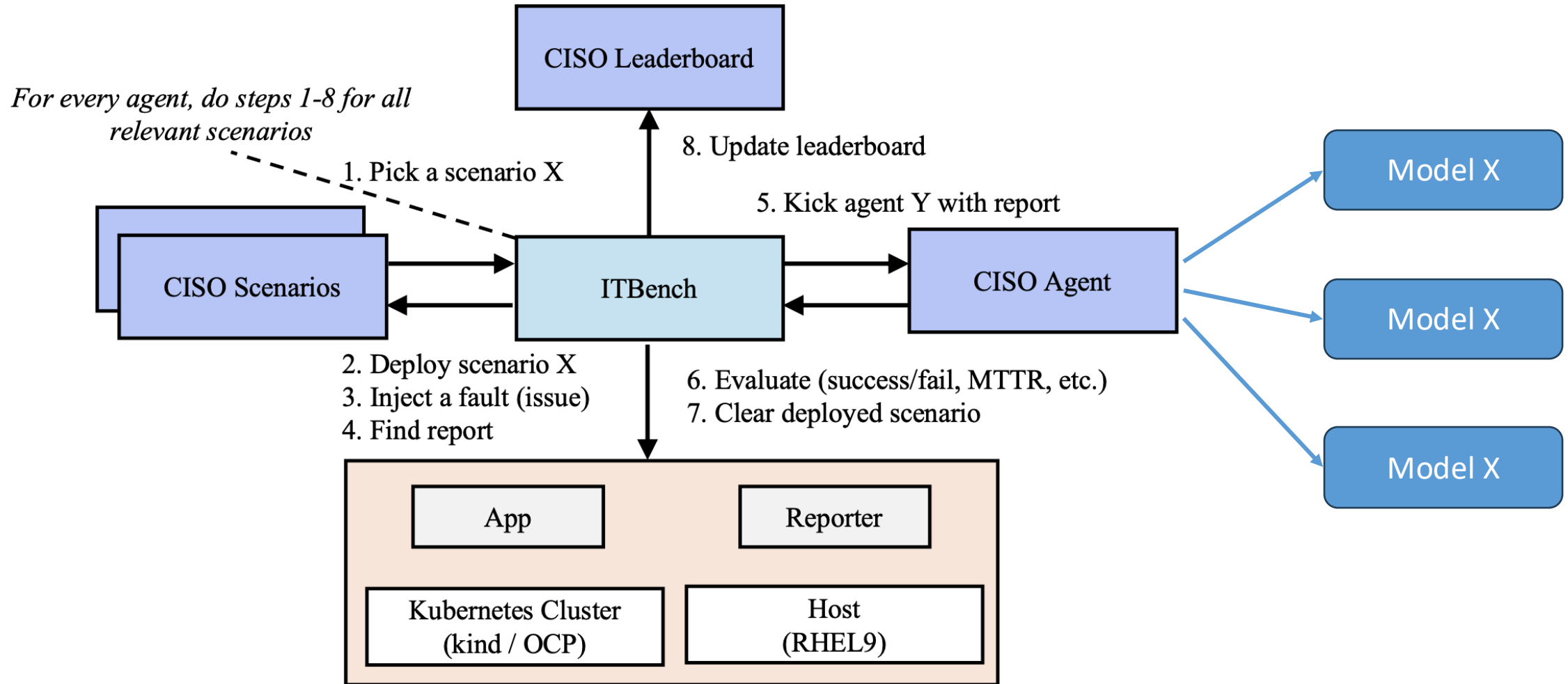


Table 5: Evaluation of CISO Compliance Assessment Agent on CISO scenarios

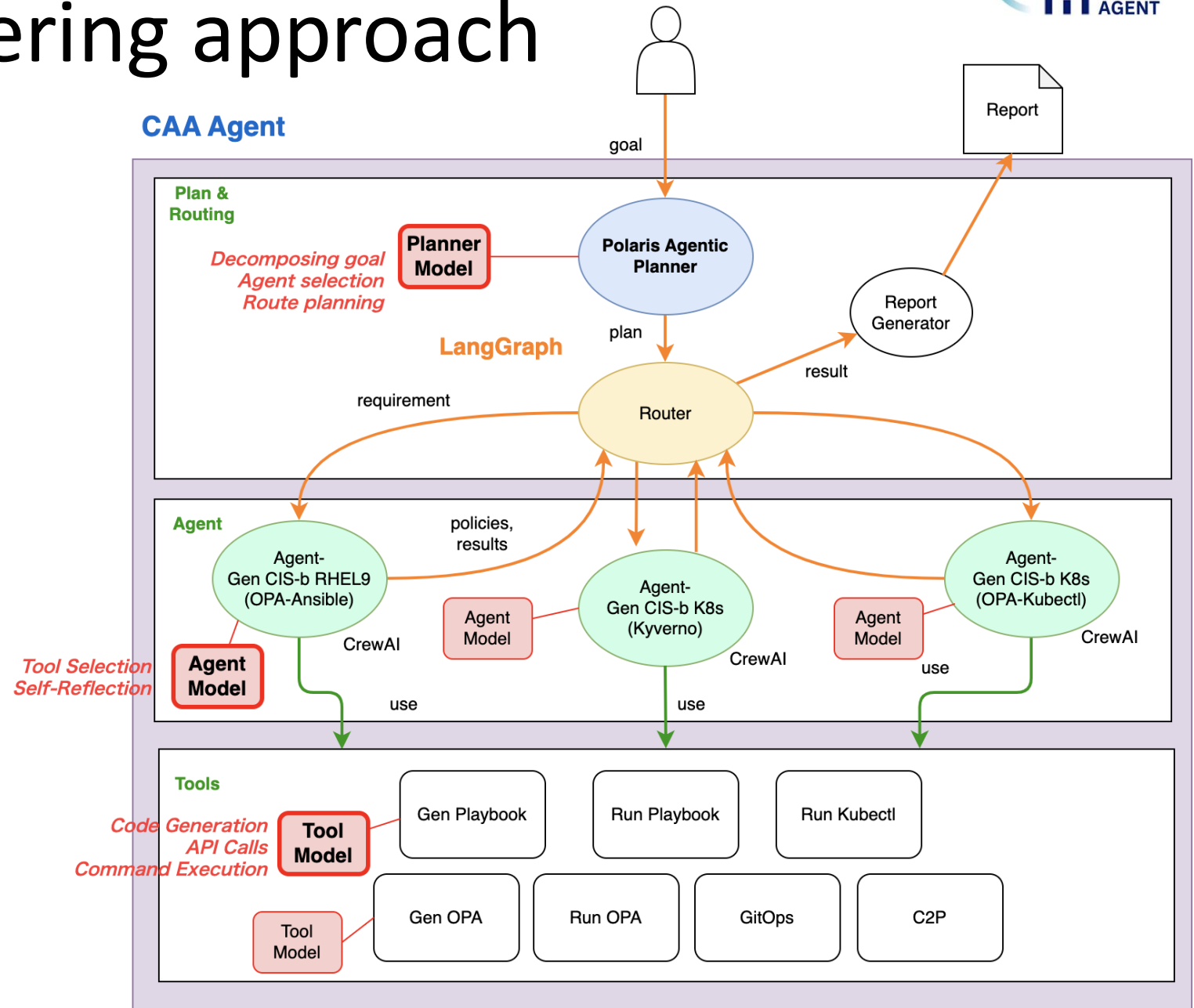
Models	Scenario pass@1 (%) \uparrow				O/A pass@1 (%) \uparrow	TTP (s) \downarrow
	kyverno	k8s-opa	rhel-opa	kyverno-update		
granite-3.1-8B-instruct	7.84 \pm 3.84	0.00 \pm 0.00	0.00 \pm 0.00	1.59 \pm 1.58	1.71 \pm 0.76	197.03 \pm 2.52
mixtral-8x7B-instruct	7.35 \pm 3.19	1.43 \pm 1.42	0.00 \pm 0.00	1.29 \pm 4.34	3.94 \pm 1.03	120.63 \pm 3.77
llama-3.1-8B-instruct	8.57 \pm 3.37	0.00 \pm 0.00	0.00 \pm 0.00	7.46 \pm 3.23	3.59 \pm 1.07	121.49 \pm 3.00
llama-3.3-70B-instruct	18.46 \pm 4.94	0.00 \pm 0.00	1.43 \pm 2.88	8.06 \pm 3.50	9.32 \pm 1.67	189.61 \pm 2.71
mistral-large-2	6.56 \pm 3.20	22.73 \pm 5.32	7.23 \pm 2.88	10.45 \pm 3.77	11.55 \pm 1.95	167.98 \pm 3.42
llama-3.1-405B-instruct	16.22 \pm 4.32	20.83 \pm 4.86	8.75 \pm 3.26	3.17 \pm 2.22	12.46 \pm 1.98	178.89 \pm 3.37
gpt-4o-mini	16.18 \pm 4.54	43.10 \pm 6.99	30.38 \pm 5.43	9.43 \pm 4.08	25.19 \pm 2.80	102.40 \pm 3.70
gpt-4o	40.28 \pm 5.99	39.34 \pm 6.55	7.61 \pm 2.81	17.74 \pm 4.92	24.74 \pm 2.64	101.29 \pm 3.81

¹ 50 scenarios. ² 8 runs per scenario per model. ³ pass@1 values are shown as percentages. ⁴ TTP Time to process (seconds).

⁵ **kyverno** = New K8s CIS-benchmarks on Kyverno, easy scenario class; **k8s-opa** = New K8s CIS-benchmarks on OPA, medium scenario class; **rhel-opa** = New RHEL9 CIS-benchmarks on Ansible-OPA, medium scenario class; **kyverno-update** = Update K8s CIS-benchmarks on Kyverno, hard scenario class.

CISO Agent – Layering approach

- Divide-and-Conquer
- Decompose high-level goal to a sequence of tasks (Polaris Agentic Planner)
- Routing to best agents (Crew) for the task (LangGraph)
- Split / minimize / structure context for performance improvement



Open Source

- CISO CAA Agent → <https://github.com/IBM/it-bench-ciso-caa-agent>
- Sample Task Scenarios → <https://github.com/IBM/itbench-sample-scenarios>

Demo

```
hiro - tmux -- tmux - 134x40
=> => exporting layers 0.0s
=> => writing image sha256:081527c77836b5bb4d5ae41ffd1fa3f6daa1c2c47ed8c1696232077694d1ea0 0.0s
=> => naming to docker.io/library/ciso-agent:latest 0.0s
it-bench-ciso-caa-agent > docker run --rm -ti --name ciso-agent \
-v /tmp/sample/agent:/tmp/agent/ \
-v /.env:/etc/ciso-agent/.env \
ciso-agent:latest \
python src/ciso_agent/main.py \
--goal "I would like to check if the following condition is satisfied, given a Kubernetes cluster w
ith `kubeconfig.yaml`
    Minimize the admission of containers wishing to share the host network namespace.
    To check the condition, do the following steps.
    - deploy a Kyverno policy to the cluster
    - check if the policy is correctly deployed.
    If deploying the policy failed and if you can fix the issue, you will do it and try deploying again.
    Once you get a final answer, you can quit the work.
    The cluster's kubeconfig is at `/tmp/agent/kubeconfig.yaml`.
    You can use `/tmp/agent` as your workdir." \
--auto-approve
=====
# Goal:
=====
I would like to check if the following condition is satisfied, given a Kubernetes cluster with `kubeconfig.yaml`
    Minimize the admission of containers wishing to share the host network namespace.
    To check the condition, do the following steps.
    - deploy a Kyverno policy to the cluster
    - check if the policy is correctly deployed.
    If deploying the policy failed and if you can fix the issue, you will do it and try deploying again.
    Once you get a final answer, you can quit the work.
    The cluster's kubeconfig is at `/tmp/agent/kubeconfig.yaml`.
    You can use `/tmp/agent` as your workdir.
[0] 0:fish 1:fish 2:fish 3:fish 4:fish 5:docker+Z "docker run --rm -ti " 13:42 25-Feb-25
```

Questions?